Physica A 421 (2015) 44–53

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

Information filtering via balanced diffusion on bipartite networks

Da-Cheng Nie [a], Ya-Hui An [a], Qiang Dong [a,b,*], Yan Fu [a,b], Tao Zhou [a,b]

[a] Web Sciences Center, School of Computer Science & Engineering, University of Electronic Science and Technology of China, Chengdu 611731, PR China

[b] Big Data Research Center, University of Electronic Science and Technology of China, Chengdu 611731, PR China

HIGHLIGHTS

- The optimal hybrid algorithm of MD and HC processes is investigated.
- BD algorithm gives recommendations with superior accuracy and diversity.
- BD algorithm recommends more unpopular objects to users.

ARTICLE INFO

Article history:
Received 5 May 2014
Received in revised form 18 September 2014
Available online 12 November 2014

Keywords:
Recommendation algorithm
Mass diffusion
Heat conduction
Bipartite network

ABSTRACT

The recent decade has witnessed the increasing popularity of recommender systems, which help users acquire relevant commodities and services from overwhelming resources on Internet. Some simple physical diffusion processes have been used to design effective recommendation algorithms for user–object bipartite networks, such as mass diffusion (MD) and heat conduction (HC) algorithms, which have different advantages respectively on accuracy and diversity. In this paper, we explore how to combine MD and HC processes to get better recommendation performance and propose a new algorithm mimicking the hybrid of MD and HC processes, named balanced diffusion (BD) algorithm. Numerical experiments on three benchmark data sets, *MovieLens*, *Netflix* and *RateYourMusic*, show that BD algorithm outperforms three typical diffusion-like algorithms on the three important metrics, accuracy, diversity and novelty. Specifically, it not only provides accurate recommendation results, but also yields higher diversity and novelty in recommendations by accurately recommending unpopular objects.

1. Introduction

Web 2.0 and its applications have achieved significant developments in the past few years, which bring us more convenience as well as overwhelm us with the information ocean on Internet [1]. This is the so-called *Information Overload* problem [2]. Nowadays, online shopping becomes more and more popular in our daily life. For instance, there are millions of books (e-books) on Amazon.com, and the turnover of Taobao.com exceeded 35 billion RMB (China's currency, about 6 billion US dollars) on the shopping festival day of Nov 11, 2013 [3]. In this case, we find that it is very difficult to choose the relevant ones from countless candidates on these e-commerce websites, and thus an automatic way that can help us to

* Corresponding author at: Web Sciences Center, School of Computer Science & Engineering, University of Electronic Science and Technology of China, Chengdu 611731, PR China.
E-mail address: dongq@uestc.edu.cn (Q. Dong).

http://dx.doi.org/10.1016/j.physa.2014.11.018

make right decision under the information overload circumstance becomes a significant issue in both academic and industrial communities.

The emergence of search engines partially alleviates this dilemma; a user inputs the keywords and then search engines return the results accordingly. However, search engines always return the same results to different users if they key in the same words. When users resort to search engines, they have already known what they want and can easily find keywords to describe it. But in the most occasions, users do not know what they really want, or it is hard to find appropriate words to describe it. Therefore, recommender systems have been designed to solve this problem. We can see that in recent years, recommender systems have greatly promoted the development of E-business, and vice versa [4].

Collaborative filtering (CF) [5–12] is the most frequently used technology in recommender systems, which makes use of the object collecting history of users to predict the potential objects of interest to the target user, including user-based CF [10] and object-based CF [8,12]. However, the original CF methods give recommendation by computing the similarity between user preferences, which will make the recommendation results more and more similar among users [13,14]. What is more, CF algorithms cannot deal with the cold start problem [15], i.e., when a new user or object is added to the system, it is difficult to obtain recommendations or to be recommended. Therefore, the content-based [16] methods have been proposed to solve this problem, which generate recommendation results by computing the similarity between user profiles, but user profiles are usually difficult to acquire due to the constraint of information retrieval techniques. Generally speaking, CF methods and content-based methods will generate similar recommendation results with poor diversity and novelty. For the recent developments of recommender systems, the readers are referred to the comprehensive review by Lü et al. [17].

To improve the diversity and novelty of recommendation results, many other personalized recommendation algorithms have been proposed, including trust-aware methods [18,19], social-impact methods [20–22] and tag-aware methods [23]. Recently, based on two physical processes, the mass diffusion (MD) algorithm and heat conduction, many effective recommendation algorithms have been designed on user–object bipartite networks, including the MD algorithm and HC algorithm [14,24,25]. The MD algorithm is essentially a resource redistribution process between objects via neighboring users [24,26], which achieves high accuracy but low diversity. Zhang et al. [27] introduce a voting system in the diffusion process to get better recommendation results. The HC algorithm is like a heat conduction process from objects to neighboring users and back to objects again, which has high diversity but low accuracy. Ideally, a good recommendation algorithm should exhibit both of high accuracy and high diversity.

In Ref. [14], Zhou et al. proposed an algorithm to nonlinearly combine the MD and HC processes (HHP for short), which solves the apparent diversity–accuracy dilemma of recommender systems. Liu et al. [28] proposed a biased heat conduction (BHC for short) algorithm, which simultaneously enhances the accuracy and diversity by decreasing temperatures of small-degree objects in the heat conduction process. Lü et al. [29] proposed a preferential diffusion algorithm, taking into account the heterogeneity of users' degrees. All of the above-mentioned algorithms derived from MD and/or HC processes demonstrate good accuracy and diversity. However, the strategy of combining MD and HC processes into one recommender system to get the optimal accuracy and diversity remains to be an open problem.

In this paper, we explore how to combine MD and HC processes to get better recommendation performance and propose a new algorithm mimicking the hybrid of MD and HC processes, named balanced diffusion (BD) algorithm. Numerical experiments on three benchmark data sets, *MovieLens*, *Netflix* and *RateYourMusic*, show that the BD algorithm outperforms three typical diffusion-like algorithms on the three important metrics, accuracy, diversity and novelty. Specifically, it not only provides accurate recommendation results, but also yields higher diversity and novelty in recommendations by accurately recommending unpopular objects.

2. Methods

A recommender system can be represented by a bipartite network $G(U, O, E)$, where $U = \{u_1, u_2, \ldots, u_m\}$, $O = \{o_1, o_2, \ldots, o_n\}$, and $E = \{e_1, e_2, \ldots, e_q\}$ represent the m users, n objects, and q links between the m users and n objects, respectively. The system could be fully described by an adjacency matrix $A = \{a_{l\alpha}\}_{m,n}$, where $a_{l\alpha} = 1$ if there exists a link $e_{l\alpha}$ between user u_l and object o_α and $a_{l\alpha} = 0$ otherwise.

We assume that a user collects an object because he/she likes it, then the essential task of a recommender system becomes to generate a ranking list of the target user's uncollected objects. All the recommendation algorithms inspired by diffusion-like process work by initially assigning all the objects a certain amount of resources, denoted by the vector $\mathbf{f}$ (where f_α is the resource of object o_α), and then reallocating these resources via the transformation $\mathbf{f}' = W\mathbf{f}$, where W is called the resource transfer matrix.

The original recommendation algorithm mimicking the mass diffusion process is called the MD algorithm, also referred to as Network-Based Inference (NBI) [24] and ProbS [14]. For a target user u_l, the initial resource vector $\mathbf{f}$ on the objects is defined as $f_\alpha = a_{l\alpha}$, where $a_{l\alpha} = 1$ if user u_l has collected object o_α, otherwise $a_{l\alpha} = 0$. The element $w_{\alpha\beta}$ of the transfer matrix W is written as

$$w_{\alpha\beta} = \frac{1}{k_{o_\beta}} \sum_{l=1}^{m} \frac{a_{l\alpha} a_{l\beta}}{k_{u_l}}, \tag{1}$$

where $k_{o_\beta} = \sum_{i=1}^{m} a_{i\beta}$ and $k_{u_l} = \sum_{r=1}^{n} a_{lr}$ denote the degrees of object o_β and user u_l, respectively.

The original recommendation algorithm mimicking mass diffusion process is called HC algorithm, also referred to as HeatS [14]. The significant difference between MD and HC is the resource redistribution strategy: MD works by equally distributing the resource of each node to its nearest neighbor, the overall resource remains unchanged; while in HC every node absorbs equal proportion of the resource from the nearest neighbors, the overall resource increases in the process. Specifically, the difference of HC from MD lies in the transfer matrix W, which is described as:

$$w_{\alpha\beta} = \frac{1}{k_{o_\alpha}} \sum_{l=1}^{m} \frac{a_{l\alpha} a_{l\beta}}{k_{u_l}}. \tag{2}$$

As we know, MD has high recommendation accuracy yet low diversity, while HC, which is designed to address the challenge of diversity, has relatively low accuracy. Many researchers attempted to solve this diversity–accuracy dilemma and have found some effective ways.

2.1. Baseline methods

Zhou et al. [14] proposed a recommendation method to nonlinearly combine MD and HC, called the HHP algorithm, by introducing a tunable parameter λ into the transfer matrix W:

$$w_{\alpha\beta} = \frac{1}{{k_{o_\alpha}}^{1-\lambda} {k_{o_\beta}}^{\lambda}} \sum_{l=1}^{m} \frac{a_{l\alpha} a_{l\beta}}{k_{u_l}}, \tag{3}$$

when $\lambda = 0$, HHP reduces to the original HC and $\lambda = 1$ the original MD. We can make a trade-off between diversity and accuracy by adjusting parameter λ.

The MD algorithm can be regarded as a three-step diffusion from the target user to final objects (i.e., user $\rightarrow$ object $\rightarrow$ user $\rightarrow$ object). Motivated by enhancing the ability to find unpopular and niche objects, Lü et al. [29] proposed a preferential diffusion method (PD for short), where a user redistributes his resource to his neighbor object o_α the amount proportional to $k_{o_\alpha}^{\varepsilon}$ in the last step, where $-1 \le \varepsilon \le 0$ is a free parameter, the resource transfer matrix reads:

$$w_{\alpha\beta} = \frac{1}{k_{o_\beta} k_{o_\alpha}^{-\varepsilon}} \sum_{l=1}^{m} \frac{a_{l\alpha} a_{l\beta}}{\mathcal{M}}, \tag{4}$$

where $\mathcal{M} = \sum_{r=1}^{n} a_{lr} k_{o_r}^{\varepsilon}$. Clearly, when $\varepsilon = 0$, it reduces to the original MD algorithm.

In Ref. [28], Liu et al. proposed a Biased Heat Conduction (BHC for short) method based on HC. By decreasing the temperatures of small-degree objects, BHC could simultaneously enhance the accuracy and diversity. The element $w_{\alpha\beta}$ of the transfer matrix W is:

$$w_{\alpha\beta} = \frac{1}{k_{o_\alpha}^{\lambda}} \sum_{l=1}^{m} \frac{a_{l\alpha} a_{l\beta}}{k_{u_l}}, \tag{5}$$

where $0 < \lambda < 1$, which indicates that the role of large-degree objects would be strengthened in the last diffusion step.

2.2. Balanced diffusion

In order to explore the optimal weights of objects of different degrees in different diffusion steps, we give two separate parameters a and b in the transfer matrix W:

$$w_{\alpha\beta} = \frac{1}{k_{o_\alpha}^{a} k_{o_\beta}^{b}} \sum_{l=1}^{m} \frac{a_{l\alpha} a_{l\beta}}{k_{u_l}}. \tag{6}$$

This equation subsums the transfer matrices of all diffusion-like algorithms, because when we adjust the parameters a and b accordingly, we will get MD, HC, BHC, PD and HHP, respectively. The HHP method is regarded as one of the best methods to combine MD and HC up to now. However, we find that if a and b are tuned appropriately to an equal value, the recommendation results will be better than HHP, BHC and PD. If $a = b > 0$, Eq. (6) can be revised using only one parameter λ:

$$w_{\alpha\beta} = \frac{1}{(k_{o_\alpha} k_{o_\beta})^{\lambda}} \sum_{l=1}^{m} \frac{a_{l\alpha} a_{l\beta}}{k_{u_l}}. \tag{7}$$

We derive from Eq. (7) the Balanced Diffusion algorithm (BD for short), where MD and HC obtain the same weights in recommender systems. That is to say, when we combine MD and HC together, we would better assign the same weight to them and put the same emphasis on objects of different degrees in diffusion steps. Therefore, we call it a balanced diffusion process. In this sense, the influence of large degree objects would be strengthened both in the second and last steps if $\lambda < 1$ and depressed if $\lambda > 1$. Comparatively speaking, the PD algorithm considers the preferential diffusion in the last step, the BHC algorithm considers the effect of object degrees in the last step, while the HHP and BD algorithms take into account the effect of object degrees both in the second and last steps.

Table 1
Properties of the tested data sets.

Data sets	Users	Objects	Links	Sparsity
MovieLens	943	1682	100,000	6.30×10^{-2}
Netflix	10,000	5640	701,947	1.24×10^{-2}
RYM	33,762	5267	675,817	3.8×10^{-3}

3. Experiments

3.1. Data set description

To test the algorithm's performance, we employ three different data sets (see Table 1 for basic statistics). The *MovieLens* data set was collected by the GroupLens Research Project at the University of Minnesota. The data was collected through the MovieLens web site (movielens.umn.edu) during the seven-month period from September 19, 1997 through April 22, 1998. The *Netflix* data set is a randomly selected subset of the huge data set provided for the Netflix Prize [30]. The *RYM* data set is obtained by downloading publicly available data from the music rating website RateYourMusic.com. We use the information of the links between users and objects in this paper.

3.2. Performance metrics

MovieLens, *Netflix* and *RYM* are three frequently-used data sets for testing the performance of recommender algorithms in the literature. Every data set is randomly divided into two parts: the training set E_T which is consisted of 90% entries and the testing set E_P consisting of the remaining 10% entries.

For a general recommendation process, the training set is treated as known information to run algorithms and generate corresponding recommendation results, and the information in the testing set which is unavailable while making recommendations is used to evaluate the results of recommendation algorithms. In this paper, we use four different metrics to carry out the evaluations in order to assess every method's all-around performance.

Accuracy is the most important measure in evaluating the performance of recommendation algorithms. A good algorithm is expected to give accurate recommendations, namely higher ability to find what users like. In order to measure the recommendation accuracy, we take ranking score r [29] and precision enhancement $ep(L)$ [14] as metrics. For a target user u_l, the recommender system will return a ranking list of all uncollected objects to him/her. For each link $e_{l\alpha}$ in the test set, we compute the rank $r_{i\alpha}$ of object o_α in the recommendation list of user u_l.

$$r_{i\alpha} = \frac{p_\alpha}{l_i} \tag{8}$$

where object o_α is listed in the p_αth position of the ranking list of user u_i, l_i is the number of uncollected objects of user u_i in the training set. The rank score r of the whole system is the average $r_{i\alpha}$ on all the entries of the testing set.

$$r = \frac{1}{|E_P|} \sum_{e_{i\alpha} \in E_P} r_{i\alpha} \tag{9}$$

where $e_{i\alpha}$ denotes the link connecting u_i and o_α in the test set.

A random recommendation will randomly choose L objects from the training data for a target user, so we consider the precision enhancement values $ep(L)$ relative to the precision of random recommendations.

$$ep(L) = \frac{1}{m} \sum_{l=1}^{m} \frac{N_{rs}^l}{L} \frac{p_l}{k_{l_{test}}} \tag{10}$$

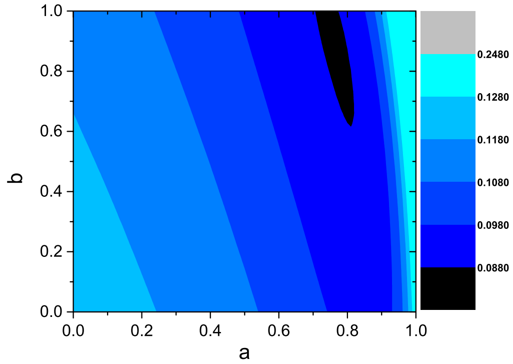
where N_{rs}^l is the number of recommended objects to user u_l truly recovered in the testing set, p_l is the number of uncollected objects of user u_l in the training sets and $k_{l_{test}}$ is the degree of user u_l in the testing set, L is the length of the recommendation list.

Beside accuracy, diversity is taken into account as another important metric to evaluate the recommendation algorithm. In order to measure the recommendation diversity and novelty, we make use of Hamming distance ($h(L)$ for short) [31] and self information $I(L)$ [14], respectively.

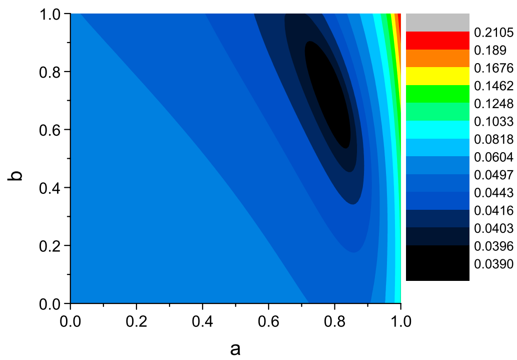
$$h_{ij}(L) = 1 - \frac{q_{ij}(L)}{L} \tag{11}$$

where $q_{ij}(L)$ is the number of common objects in the top L places of both recommendation lists of user u_i and user u_j. Averaging $h_{ij}(L)$ over all pairs of users existing in the test set, we obtain the average hamming distance $h(L)$, for which the greater value means greater personalization of users' recommendation lists.

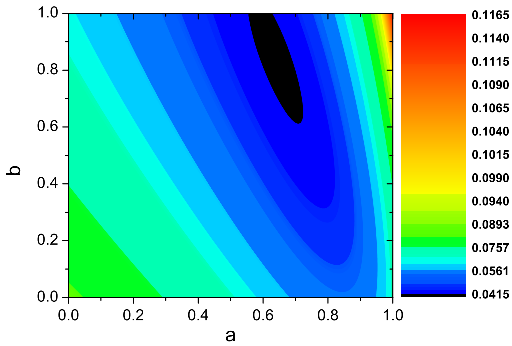
The $I(L)$ concerns the capacity of the recommender system to generate novel and unexpected results. Given an object o_α, the chance a randomly selected user has collected it is k_{o_α}/m, and its self-information is defined as $I_{o_\alpha} = \log_2(m/k_{o_\alpha})$. The



(a) MovieLens.



(b) Netflix.



(c) RYM.

Fig. 1. The ranking scores on *MovieLens*, *Netflix* and *RYM* data sets according to Eq. (6). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

self-information $I(L)$ of the whole system is:

$$I(L) = \sum_{l=1}^{m} \sum_{\alpha=1}^{L} I_{o_\alpha}. \quad (12)$$

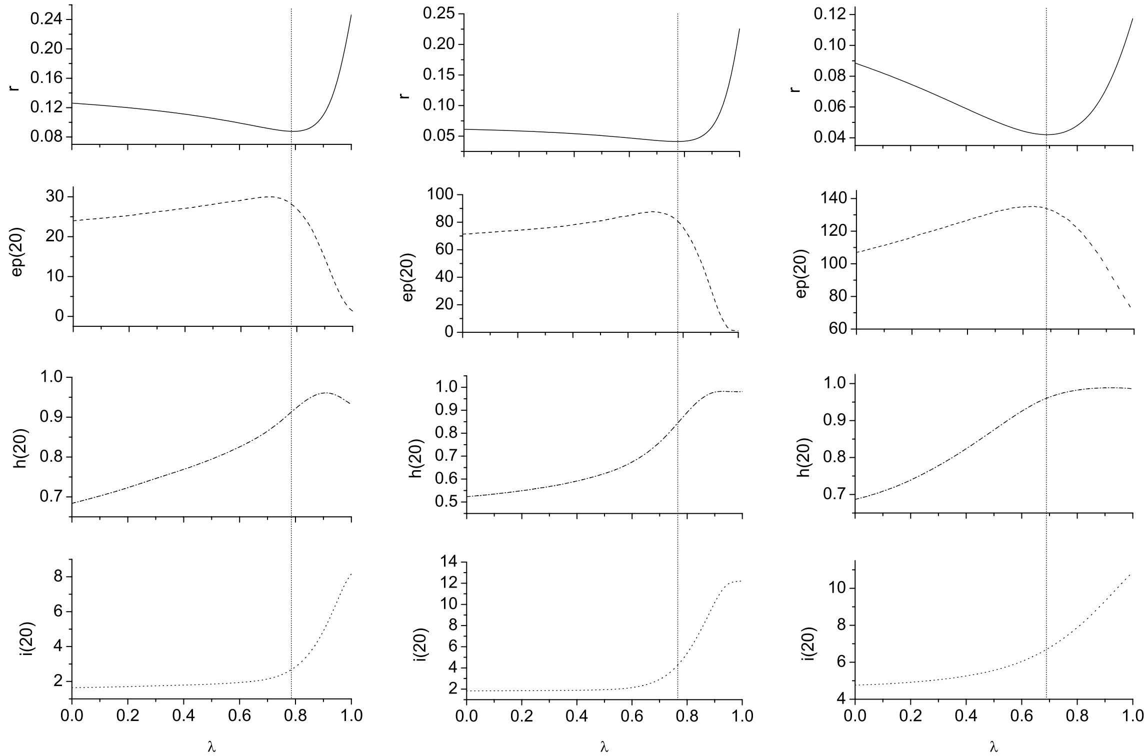
3.3. Experimental results

Fig. 1 shows the ranking score values by Eq. (6) with various a and b parameters on three benchmark data sets, *MovieLens*, *Netflix* and *RYM*, respectively. In the figure, different colors represent different r. The ranking score in the black area falls in the range of 0.087–0.088, 0.039–0.040, and 0.041–0.0415 respectively on *MovieLens*, *Netflix* and *RYM* data sets. The ranking score value remains almost unchanged when we adjust the two parameters within the black area. Thus, we can choose any point in the black area as the optimal parameter settings giving the smallest ranking score.

In the BHC algorithm, the influence of large-degree objects is strengthen in the last diffusion step [28]. Nie et al. [32] take into account the effect of object degree in the second step of the standard mass diffusion algorithm, which shows that depressing the influence of large-degree objects in the second diffusion step will simultaneously enhance the accuracy and diversity. Intuitively, when we combine MD and HC into recommender systems we should depress the influence of large-degree objects in the second diffusion step and strengthen the influence of large-degree objects in the last diffusion step. In other words, when combine MD and HC into recommender systems the parameters of Eq. (6) should satisfy $b > 1$ and $0 < a < 1$. However, in every subfigures of Fig. 1, the diagonal line always goes through the black area, without loss of generality, we can simply take $0 < a = b < 1$ for the transfer matrix of a novel recommendation algorithm. Therefore, we could only use one parameter λ to replace the two in the optimal scenario (i.e., Eq. (7)).

Fig. 2 shows the performance of BD algorithm on *MovieLens*, *Netflix* and *RYM*, respectively. The *optimal parameter* λ is determined by the smallest ranking score, which are 0.79, 0.77 and 0.69 on *MovieLens*, *Netflix* and *RYM* data sets, respectively. The other three metrics, precision, hamming distance and novelty's optimal values are obtained at the point of optimal λ. From the figure, we can see that BD is a good trade-off of the diversity and accuracy. In other words, our algorithm exhibits outstanding diversity and accuracy simultaneously.

Generally speaking, the MD algorithm gives high recommendation accuracy yet low diversity, and the HC algorithm gives high recommendation diversity but low accuracy. The HC algorithm has a bias on the unpopular objects, but the power-law



(a) MovieLens. (b) Netflix. (c) RYM.

Fig. 2. Performance of the BD algorithm on three different data sets.

object degree distribution indicates that large-degree objects are preferred by many users [28]. Therefore, in order to obtain more accurate recommendation, the large-degree objects should be paid more attention in the last diffusion step. In contrast, the MD algorithm has a bias on the popular objects; therefore, the small degree objects are difficult to be recommended, and the role of large-degree objects should be depressed in the second diffusion step to obtain the diverse recommendation. However, the optimal parameter of BD algorithm is $\lambda < 1$, which indicates that when we combine the MD and HC processes, the role of large-degree objects would be strengthened both in the second and last diffusion steps to obtain more accurate and diverse recommendation.

In order to explain why we called our algorithm balanced diffusion and why both the accuracy and diversity are enhanced by the BD algorithm, Fig. 3(b)–(f) count the number $N(k)$ of objects of degree k in all users' recommendation lists respectively given by HC, MD, Biased Random Walk (BRW) [32], BHC and BD algorithms on Netflix data set, where the recommendation length is $L = 20$. Fig. 3(a) shows that the object degree distribution on the training set follows power-law distribution.

Fig. 3(b)–(c) show that the HC algorithm tends to recommend the small-degree objects, while the MD algorithm large-degree objects. Fig. 3(d)–(f) show the object number $N(k)$ of degree k against the object degree k at the optimal parameter point (say optimal λ), respectively. We can see that the points on the 2-dimensional plane for the BRW, BHC and BD algorithms all form butterfly shapes. The points on two wings of the butterfly represent the small-degree objects and the large-degree objects in the recommendation lists.

However, these three algorithms have different abilities to recommend large-degree objects and small-degree objects. The BD algorithm has better diversity (ability to recommend small-degree objects) than BRW and BHC. For the BD algorithm, the body of the butterfly (the break point separating the group of small-degree objects and the group of large-degree objects) lies in the average object degree $\bar{k}_o = 105$ of the training set, which is not true for BRW and BHC algorithms. Therefore, BD balances the group of large-degree and the group of small-degree objects in the recommendation list.

In order to show BD algorithm's superior performance, we compare BD with HHP, BHC and PD in the four metrics: ranking score r, precision enhancement $ep(L)$, Hamming distance $h(20)$ and self-information $I(20)$ in Figs. 4–6. To compare the four algorithms in the same scale, we use λ instead of $1 - \lambda$ and $-\varepsilon$ for HHP and PD algorithms, respectively. Figs. 4–6 show the performance of the four algorithms with different λ on *MovieLens*, *Netflix* and *RYM* data sets, respectively.

The metric values of the four algorithms with optimal λ on *MovieLens*, *Netflix* and *RYM* data sets are shown in Tables 2–4, respectively. Clearly, BD outperforms HHP and BHC on the *MovieLens* data set, outperforms BHC and PD on the *Netflix* data set, and outperforms HHP and PD on the *RYM* data set, where "outperform" means "be better in all the four metrics". Among all these four algorithms, BD gives the best ranking score and hamming distance.

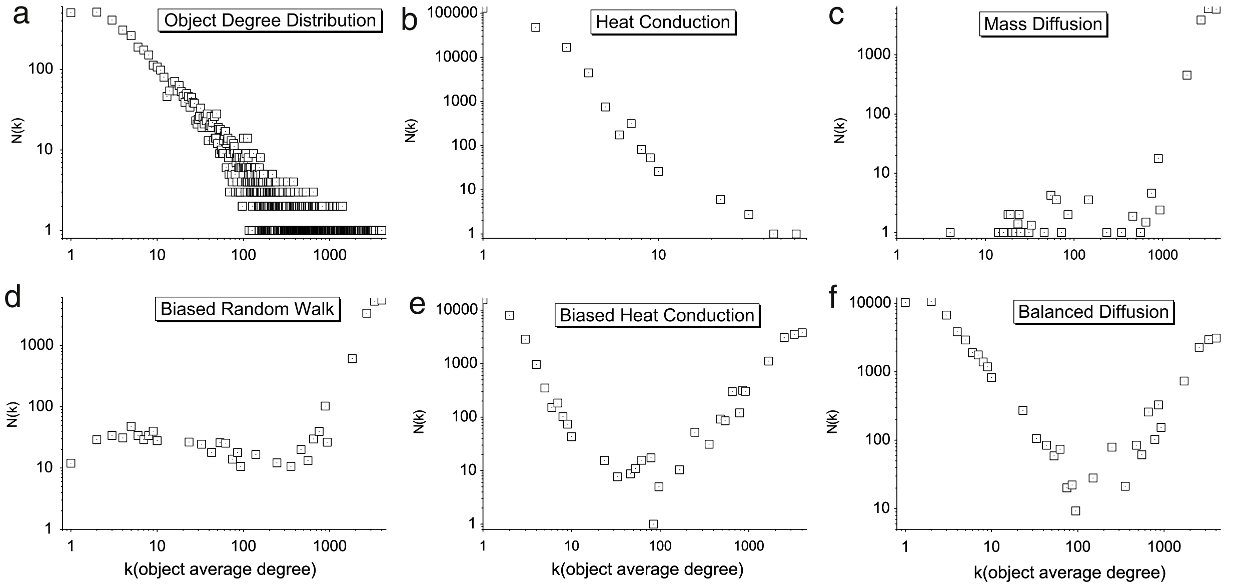


Fig. 3. (a) Object degree distribution of Netflix data set and (b)–(f) correlations between the occurrence number $N(k)$ and the object degree k of HC, MD, BRW, BHC and BD algorithms respectively with $L = 20$. The results for MovieLens and RYM are similar.

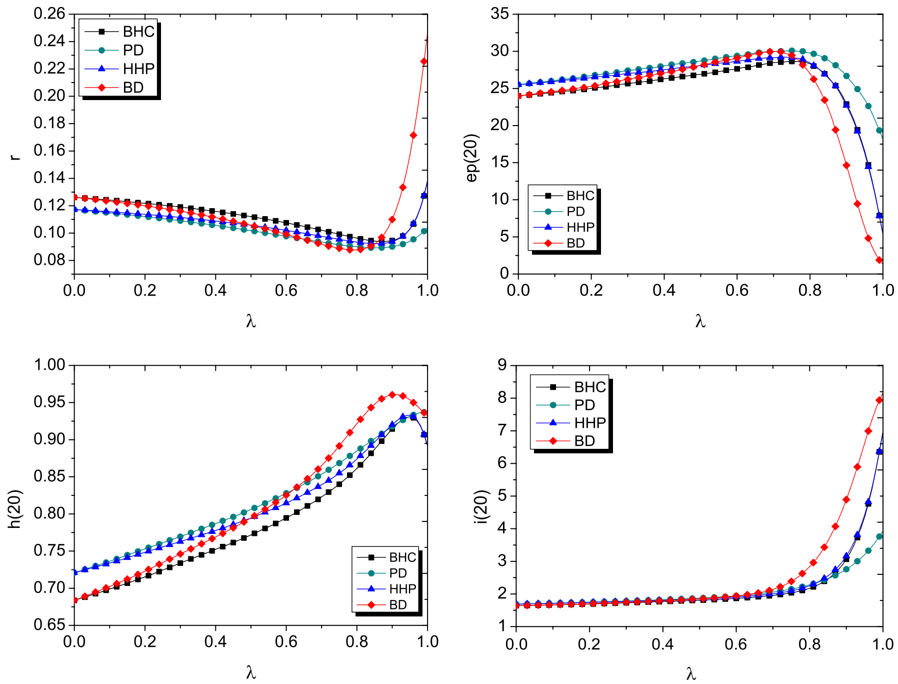


Fig. 4. (Color online) The recommendation results of four algorithms on the MovieLens data set.

Generally speaking, BD gives recommendation results with almost the best accuracy and diversity. For instance, on the *MovieLens* data set, the BD algorithm decreases the ranking score to 0.08769 while simultaneously improves the hamming distance and novelty to 0.91572 and 2.7269. Meanwhile, the BD algorithm's precision enhancement is 27.63, which is very close to the best value 28.793 of PD, much better than those of HHP and BHC. In other words, experimental results show that the balanced diffusion both at the second and last diffusion steps is effective in simultaneously recommending popular and unpopular objects.

In order to further explain why the BD algorithm has better ability of recommending small-degree objects, we plot the ranking scores against the object degree on *MovieLens*, *Netflix* and *RYM* data sets in Fig. 7. For a given x, its corresponding r is obtained by averaging over all objects whose degrees are in the range of $[a(x^2 - x), a(x^2 + x)]$, where a is chosen as $\frac{1}{2}\log 5$ for a better illustration.

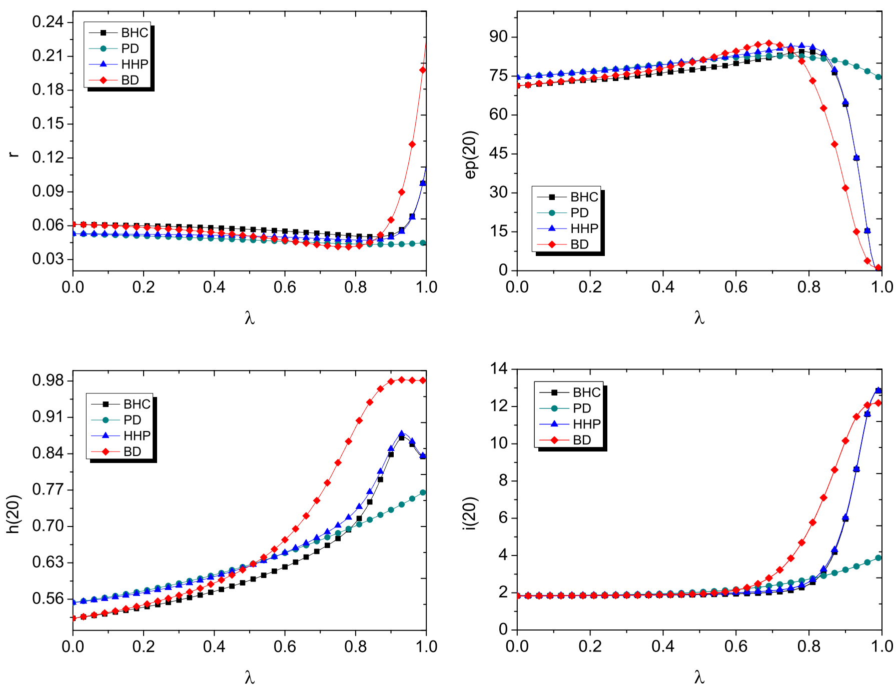


Fig. 5. (Color online) The recommendation results of four algorithms on the Netflix data set.

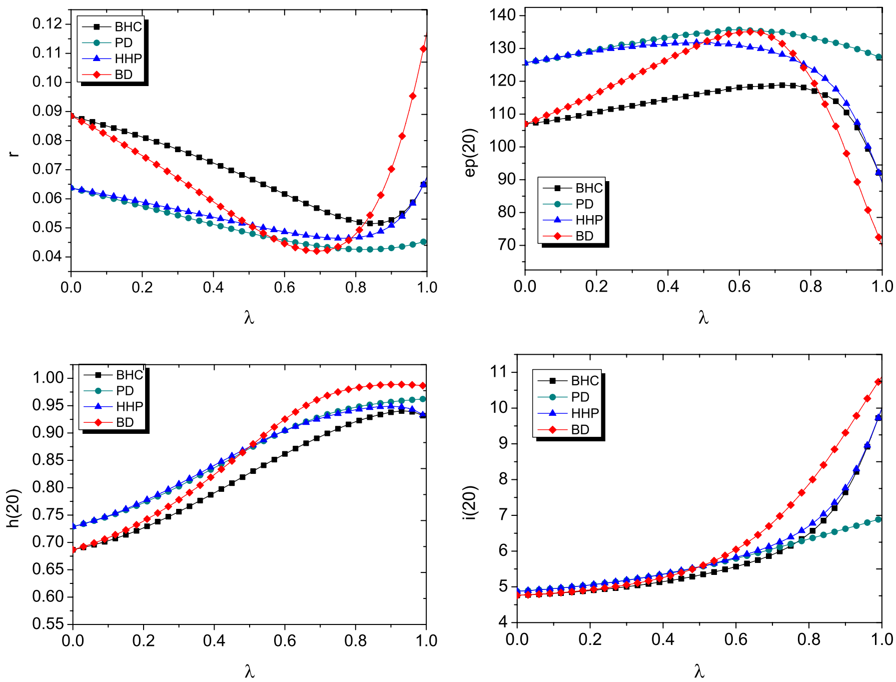


Fig. 6. (Color online) The recommendation results of four algorithms on the RYM data set.

Table 2
Algorithmic performance for the MovieLens data set. The optimal parameters are $\lambda_{opt} = 0.14$ for HHP, $\lambda_{opt} = 0.87$ for BHC, $\varepsilon_{opt} = -0.85$ for PD and $\lambda_{opt} = 0.79$ for BD.

Method	r	$ep(20)$	$h(20)$	$I(20)$
HHP	0.09228	25.892	0.90162	2.6452
BHC	0.09388	25.367	0.89809	2.6474
PD	0.08924	**28.793**	0.90146	2.4716
BD	**0.08769**	27.63	**0.91572**	**2.7269**

Table 3
Algorithmic performance for the Netflix data set. The optimal parameters are $\lambda_{opt} = 0.17$ for HHP, $\lambda_{opt} = 0.85$ for BHC, $\varepsilon_{opt} = -0.88$ for PD and $\lambda_{opt} = 0.77$ for BD.

Method	r	$ep(20)$	$h(20)$	$I(20)$
HHP	0.04719	**84.89511**	0.75589	3.03893
BHC	0.05023	81.02979	0.76022	3.41337
PD	0.04348	80.68701	0.72574	3.1171
BD	**0.04125**	82.53697	**0.85025**	**4.38952**

Table 4
Algorithmic performance for the RYM data set. The optimal parameters are $\lambda_{opt} = 0.24$ for HHP, $\lambda_{opt} = 0.85$ for BHC, $\varepsilon_{opt} = -0.82$ for PD and $\lambda_{opt} = 0.69$ for BD.

Method	r	$ep(20)$	$h(20)$	$I(20)$
HHP	0.04642	126.15419	0.93689	6.44823
BHC	0.05151	115.28652	0.93062	**6.95915**
PD	0.04259	122.52548	0.94475	6.85662
BD	**0.04202**	**133.46272**	**0.9605**	6.6999

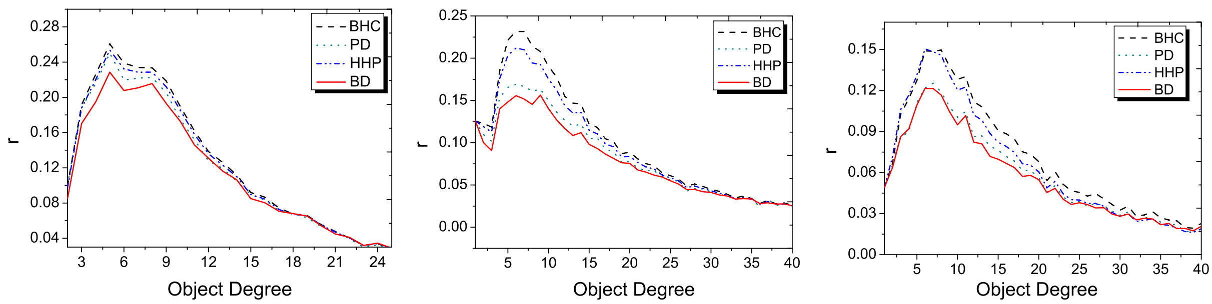


Fig. 7. (Color online) Dependence of ranking score r on the object degree. For a given x, its corresponding r is obtained by averaging over all objects whose degrees are in the range of $[a(x^2 - x), a(x^2 + x)]$, where a is chosen as $\frac{1}{2}\log 5$ for a better illustration.

From the figure, we can see that the ranking score values of popular objects are much smaller than those of unpopular objects. That is to say, the popular objects have more opportunity to be recommended and can be more accurately recommended. Specifically, for the objects of the same degree, the ranking score value of BD algorithm is always smaller than those of HHP, BHC and PD algorithms, especially for small-degree objects. Thus, the BD algorithm has the best performance on recommending unpopular objects among the four algorithms.

4. Conclusion and discussion

In this paper, we investigate how to combine the mass diffusion (MD) and heat conduction (HC) processes into one algorithm, in order to get the optimal recommendation performance. Experimental results show that the smallest ranking score value occurs when MD and HC are assigned almost the same weight in the hybrid algorithm. Thus, we proposed the Balanced Diffusion (BD) algorithm, which simultaneously improves the accuracy and diversity of recommender systems. The BD algorithm outperforms other three typical diffusion-based algorithms in almost four metrics: ranking score, precision enhancement, Hamming distance and self-information, on three benchmark data sets. Specially, the BD algorithm has much better ability to recommend unpopular objects compared to the other three algorithms.

In a real online recommender system, large-degree objects are popular with users and easily to be recommended, while small-degree objects might be difficult to be recommended. Therefore, the ability to accurately recommend the unpopular objects is an important issue in recommender systems. In other words, this work provides a practical solution for online recommendation on how to promote the users' attention on the long-tailed products.

This paper provides a simple method to combine the mass diffusion and heat conduction by giving them the same weight, while a couple of issues remain open for future study. First, we have little knowledge of quantitative analysis of the structure and dynamics of information network. Second, the role of social network is overlooked in recommendation systems. Although the interaction between social influence and recommender systems is not clear thus far, we deem that an in-depth understanding of social network will be helpful for better recommendations. Finally, the multi-layered network consists of social network and the information network can be taken into account to describe the underlying hierarchical structure; thus the Social Network Analysis (SNA)-based techniques can be used to provide more substantial recommendations, and social predictions as well.

Acknowledgments

The authors would like to express their gratitude to Zi-Ke Zhang, Jun-Lin Zhou and Xu-Zhen Zhu for helpful discussions and irradiative ideas. This work is partially supported by National Natural Science Foundation of China (Grant Nos. 61103109, 11105024 and 61300018), Special Project of Sichuan Youth Science and Technology Innovation Research Team (Grant No. 2013TD0006), Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20120185120017), Fundamental Research Funds for the Central Universities (No. ZYGX2012J071), and China Postdoctoral Science Foundation (Grant Nos. 2013M531951 and 2014T70860).

References

[1] A.M. Froomkin, Flood control on the information ocean: living with anonymity, digital cash, and distributed databases, J. Law Commer. 15 (1995) 395.
[2] A. Edmunds, A. Morris, The problem of information overload in business organisations: a review of the literature, Int. J. Inf. Manage. 20 (2000) 17–28.
[3] H. Liu, China sets online shopping record on Singles' day, 2013.
[4] M.B. Dias, D. Locher, M. Li, W. El-Deredy, P.J. Lisboa, The value of personalised recommender systems to e-business: a case study, in: Proceedings of the 2008 ACM Conference on Recommender Systems, ACM, 2008, pp. 291–294.
[5] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Trans. Knowl. Data Eng. 17 (2005) 734–749.
[6] J.B. Schafer, D. Frankowski, J. Herlocker, S. Sen, Collaborative filtering recommender systems, in: The Adaptive Web, Springer, 2007, pp. 291–324.
[7] J.L. Herlocker, J.A. Konstan, A. Borchers, J. Riedl, An algorithmic framework for performing collaborative filtering, in: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 1999, pp. 230–237.
[8] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, in: Proceedings of the 10th International Conference on World Wide Web, ACM, 2001, pp. 285–295.
[9] M. Deshpande, G. Karypis, Item-based top-N recommendation algorithms, ACM Trans. Inf. Syst. (TOIS) 22 (2004) 143–177.
[10] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl, GroupLens: an open architecture for collaborative filtering of netnews, in: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, ACM, 1994, pp. 175–186.
[11] J.S. Breese, D. Heckerman, C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, in: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc., 1998, pp. 43–52.
[12] G. Linden, B. Smith, J. York, Amazon. com recommendations: item-to-item collaborative filtering, IEEE Internet Comput. 7 (2003) 76–80.
[13] W. Zeng, M.-S. Shang, Q.-M. Zhang, L. Lue, T. Zhou, Can dissimilar users contribute to accuracy and diversity of personalized recommendation? Internat. J. Modern Phys. C 21 (2010) 1217–1227.
[14] T. Zhou, Z. Kuscsik, J.-G. Liu, M. Medo, J.R. Wakeling, Y.-C. Zhang, Solving the apparent diversity-accuracy dilemma of recommender systems, Proc. Natl. Acad. Sci. 107 (2010) 4511–4515.
[15] A.I. Schein, A. Popescul, L.H. Ungar, D.M. Pennock, Methods and metrics for cold-start recommendations, in: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2002, pp. 253–260.
[16] M.J. Pazzani, D. Billsus, Content-based recommendation systems, in: The Adaptive Web, Springer, 2007, pp. 325–341.
[17] L. Lü, M. Medo, C.H. Yeung, Y.-C. Zhang, Z.-K. Zhang, T. Zhou, Recommender systems, Phys. Rep. 519 (2012) 1–49.
[18] R. Burke, Hybrid Web recommender systems, in: The Adaptive Web, Springer, 2007, pp. 377–408.
[19] C. Palmisano, A. Tuzhilin, M. Gorgoglione, Using context to improve predictive modeling of customers in personalization applications, IEEE Trans. Knowl. Data Eng. 20 (2008) 1535–1549.
[20] D.-C. Nie, M.-J. Ding, Y. Fu, J.-L. Zhou, Z.-K. Zhang, Social interest for user selecting items in recommender systems, Internat. J. Modern Phys. C 24 (2013) 1350022.
[21] D.-C. Nie, Z.-K. Zhang, J.-L. Zhou, Y. Fu, K. Zhang, Information filtering on coupled social networks, PLoS One 9 (2014) e101675.
[22] D.-C. Nie, Z.-K. Zhang, Q. Dong, C. Sun, Y. Fu, Information filtering via biased random walk on coupled social network, Sci. World J. 2014 (2014) 829137.
[23] Z.-K. Zhang, T. Zhou, Y.-C. Zhang, Tag-aware recommender systems: a state-of-the-art survey, J. Comput. Sci. Tech. 26 (2011) 767–777.
[24] T. Zhou, J. Ren, M. Medo, Y.-C. Zhang, Bipartite network projection and personal recommendation, Phys. Rev. E 76 (2007) 046115.
[25] Y.-C. Zhang, M. Blattner, Y.-K. Yu, Heat conduction process on community networks as a recommendation model, Phys. Rev. Lett. 99 (2007) 154301.
[26] Z. Huang, H. Chen, D. Zeng, Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering, ACM Trans. Inf. Syst. (TOIS) 22 (2004) 116–142.
[27] Y.-C. Zhang, M. Medo, J. Ren, T. Zhou, T. Li, F. Yang, Recommendation model based on opinion diffusion, Europhys. Lett. 80 (2007) 68003.
[28] J.-G. Liu, T. Zhou, Q. Guo, Information filtering via biased heat conduction, Phys. Rev. E 84 (2011) 037101.
[29] L. Lü, W. Liu, Information filtering via preferential diffusion, Phys. Rev. E 83 (2011) 066119.
[30] J. Bennett, S. Lanning, The netflix prize, in: Proceedings of KDD Cup and Workshop, Vol. 2007, p. 35.
[31] T. Zhou, L.-L. Jiang, R.-Q. Su, Y.-C. Zhang, Effect of initial configuration on network-based recommendation, Europhys. Lett. 81 (2008) 58004.
[32] D.-C. Nie, Y. Fu, J.-L. Zhou, Z. Liu, Z.-K. Zhang, C. Liu, A personalized recommendation algorithm via biased random walk, in: 2014 11th International Joint Conference on Computer Science and Software Engineering, JCSSE, IEEE, 2014, pp. 292–296.